HP StorageWorks Replication Solutions Manager Command Line User Interface reference guide

HP StorageWorks Replication Solutions Manager Command Line User Interface reference guide

Contents

Tables

Preface

About this guide

This guide describes procedures for installing, configuring, and managing HP StorageWorks Replication Solutions Manager Command Line User Interface (hereafter called the HP CLUI).

Topics include:

- Installing HP CLUI
- Configuring HP CLUI
- Accessing HP CLUI
- Command descriptions
- BC 2.x EVMCL to RSM 1.x job-related commands

Intended audience

This guide is intended for network and storage administrators and HP-authorized service providers who are installing, configuring, or maintaining the software.

Prerequisites

Using this guide requires basic knowledge of:

- Storage Area Networks (SANs)
- SAN fabrics
- HP StorageWorks Enterprise Virtual Array
- Operating systems in your EVA and EVA management configuration
- HP StorageWorks Command View EVA

Related documentation

The following documents provide additional information about this and related products:

- *HP StorageWorks Business Copy EVA administrator guide*
- *HP StorageWorks Continuous Access EVA administrator guide*
- *HP StorageWorks Continuous Access EVA 2.1 release notes*
- *HP StorageWorks EVA software compatibility reference*
- *HP StorageWorks JREserver installation instructions*
- *HP StorageWorks Replication Solutions Manager Command Line User Interface reference guide*
- *HP StorageWorks Replication Solutions Manager online help and user guide*
- *HP StorageWorks Replication Solutions Manager 1.1 release notes*

You can find these documents on the following HP web sites:

- http://h18006.www1.hp.com/products/storage/software/conaccesseva/index.html
- http://h18006.www1.hp.com/products/storage/arraysystems.html
- http://h18006.www1.hp.com/products/storage/software/bizcopyeva/index.html

Document conventions

Table 1 Document conventions

Convention	Element
Blue text: Figure 1	Cross-reference links and e-mail addresses
Blue, underlined text: http://www.hp.com	Web site addresses
Bold font	GUI elements that are clicked or selected, such as menu and list items, buttons, and check boxes.
Italics font	Text emphasis
Monospace font	• File and directory names • System output • Code • Text typed at the command-line
Monospace, italic font	• Code variables • Command-line variables
Monospace, bold font	Emphasis of file and directory names, system output, code, and text typed at the command-line

△ CAUTION:
Indicates that failure to follow directions could result in damage to equipment or data.

NOTE:
Provides additional information.

HP technical support

Telephone numbers for worldwide technical support are listed on the following HP web site: http://www.hp.com/support/.

Collect the following information before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

For continuous quality improvement, calls may be recorded or monitored.

HP strongly recommends that customers sign up online using the subscriber's choice web site: http://www.hp.com/go/e-updates.

- Subscribing to this service provides you with e-mail updates on the latest product enhancements, newest versions of drivers, and firmware documentation updates as well as instant access to numerous other product resources.
- After signing up, you can quickly locate your products by selecting **Business support** and then **Storage** under Product Category.

Providing feedback

To provide e-mail feedback on:

- HP Command View EVA: CVfeedback@hp.com
- HP Business Copy EVA: BCfeedback@hp.com
- HP Continuous Access EVA: CAfeedback@hp.com

HP-authorized reseller

For the name of your nearest HP-authorized reseller:

- In the United States, call 1-800-345-1518.
- Elsewhere, visit http://www.hp.com and click **Contact HP** to find locations and telephone numbers.

HP storage web site

The HP web site has the latest information on this product as well as the latest drivers. Access storage from http://www.hp.com/country/us/eng/prodserv/storage.html. From this web site, select the appropriate product or solution.

Helpful web sites

For other product information, see the following web sites:

- http://www.hp.com
- http://www.hp.com/go/storage
- http://www.hp.com/support
- http://www.docs.hp.com

1 About the Replication Solutions Manager Command Line User Interface

HP StorageWorks Replication Solutions Manager is a single, centralized management tool that simplifies and automates the use of local and remote replication features on supported storage arrays. The interface, consisting of a graphical user interface and a Command Line User Interface (CLUI), provides consistency across a variety of supported arrays.

The CLUI allows you to perform various local and remote replication tasks, using individual commands and command scripts.

Installing the CLUI

The CLUI is installed when you install the Replication Solutions Manager software. See the *HP StorageWorks Replication Solutions Manager installation and administrator guide* for more information.

Configuring the CLUI

The CLUI is configured using the GUI. See the *HP StorageWorks Replication Solutions Manager online help* and *user guide* for more information.

Accessing the CLUI

You can access the CLUI by the following methods:

- Telnet
- Command Line User Interface window in the GUI
- User-written client

Using a Telnet session

1. Open a command window.

2. Type **telnet <management_server_name> <port_number>**. Press **Enter**.

   The default port number is 9000.

   A login prompt appears.

3. Log in with the username and password for the Replication Solutions Manager.

   The Command Line User Interface prompt appears. You can enter CLUI commands or scripts using the Perl Telnet package to create a scriptable client interface. The CLUI displays all responses as text.

Using the GUI

1. Launch the GUI.

2. Click **Tools > Command Line User Interface**.

   The Command Line User Interface window appears. You can enter CLUI commands in the text box. The window displays all results in the lower area.

Using a user-written client

You can create a user-written client to access the CLUI. If you create a client, it must provide user authentication and issue valid commands. You can use Perl, Java, or other programming languages to create a client. You can use the standard Telnet configured port (9000 by default) to connect with a standard socket connection, using the `login` command as the initial command. The SSL is accessed via an anonymous SSL socket using the same login method.

Sample Telnet client using Perl

```
use strict;
use Net::Telnet ();

my ($hostname, $line, $passwd, $pop, $username, $cmd, $res, $telnet, $port, $prompt, $ mode);



if(@ARGV < 5){
     usage();
 die "\nincorrect number of arguments\n\n";
}

($cmd) = @ARGV[4];
$hostname = $ARGV[0];
$port = $ARGV[1];
$username = $ARGV[2];
$passwd = $ARGV[3];

$mode = 1;

$telnet = new Net::Telnet (Telnetmode => 1,
      Timeout => 600,
   Cmd_remove_mode => $mode,
   Prompt => '/[>] $/');
$telnet->open(Host => $hostname,
       Port => $port);

## Read connection message.
$line = stripLine();

## Send user name.
$telnet->print("$username");
## Send password.
$telnet->print("$passwd");

#read the responses up to this point (ignoring)
readResponse();

#Send the command passed in as an arg
$telnet->print("$cmd");
#remove the command echo
stripLine();
```

```
#get and show the response
$res=readResponse();
print("$res\n");


exit;



sub readResponse {
 my $resp = "";
 $line = "";
 while((index($line, ">")<0) and (index($line,"Thank you for using")<0) ){
       $line = $telnet->get;
        $resp = $resp . $line;
       }
      return $resp;
}



sub stripLine{
 $telnet->getline;
}

sub usage {

 print "\n\n\********************************************************************************\n\n";
 print "Use: clui_telnet_sample.pl <host> <port> <username> <password> <\"command\">\n\n";
print "          host - ip or name\n";
 print "          port - the port the clui is on\n";
 print "          username - admin user name\n";
 print "          password - admin password\n";
      print "          command - the command to send via the clui - in quotes\n\n";
      print "********************************************************************************\n";

} # end sub usage
```

Sample Socket client using Perl

```
#Copyright: Copyright (c) 2003
#Company: Hewlett-Packard Company

use strict;
use IO::Socket;

my ($hostname, $line, $passwd, $username, $res, $sock, $port, $cmd);

if(@ARGV < 5){
     usage();
 die "\nincorrect number of arguments\n\n";
}

$hostname = $ARGV[0];
$port = $ARGV[1];
$username = $ARGV[2];
$passwd = $ARGV[3];
$cmd = $ARGV[4];

#may want to jump timeout if slow connection or remote server
#  blocks for longer than timeout val when zipping server files
$sock = new IO::Socket::INET (
                                PeerAddr => $hostname,
                                PeerPort => $port,
                                Proto => 'tcp',
                                Timeout => 60
                             );
  die "Could not create socket: $!\n" unless $sock;
```

```perl
    print $sock "LOGIN USERNAME=$username PASSWORD=$passwd\r\n";
    #read the telnet login handshake and disgard
    readResponse();

    #Send command passed in via arg
    $sock->print($cmd . "\r\n");

    $line = readResponse();
    print("$line\n");

    #close our socket
    $sock->close();

    #exit success if we make it here
    exit 0;

    #
    sub getResponse{
     #WARNING - this will block if line is not available
     my $ret = "";
     $ret = readline $sock;
     return $ret;
    }

    #sub to find the </commandresponse> string that is found
    #after setting result type to xml
    sub readResponse {
     my $resp = "";
     my $buff = '';

     $line = "";
     while(index($resp, "</commandresponse>") < 0){
         recv($sock, $buff, 1024, 0);
         $line = unpack("a1024", $buff);
         $resp = $resp . $line;
         #print("$line\n");
        }
        return $resp;
    }

    sub usage {

     print "\n\n\********************************************************************************\n\n";
     print "Use: clui_socket_config_retrieval.pl <host> <port> <username> <password> <command>\n\n";
    print "        host - ip or name\n";
     print "         port - the port the clui is on\n";
     print "          username - admin user name\n";
     print "         password - admin password\n";
         print "           command - The command to send to the server (in quotes)\n\n";
         print "********************************************************************************\n";

    } # end sub usage
```

Sample SSL client using Perl

```perl
    # a test client for testing CLUI Result Code Retrieval
    #


    use strict;
    use IO::Socket::SSL;

    my ($hostname, $line, $passwd, $username, $res, $sock, $port, $cmd);

    if(@ARGV < 5){
        usage();
```

```perl
    die "\nincorrect number of arguments\n\n";
}

$hostname = $ARGV[0];
$port = $ARGV[1];
$username = $ARGV[2];
$passwd = $ARGV[3];
$cmd = $ARGV[4];


if(!($sock = IO::Socket::SSL->new( PeerAddr => $hostname,
        PeerPort => $port,
        Proto    => 'tcp',
        SSL_use_cert => 0,
    ))) {
    warn "unable to create socket: ", &IO::Socket::SSL::errstr, "\n";
    exit(1);
}

print $sock "LOGIN USERNAME=$username PASSWORD=$passwd\r\n";
#read login handshake and disgard
readResponse();

print $sock "sel sys LA";
readResponse();
print $sock "sel man cueball";
readResponse();

#Send command passed in via arg
$sock->print($cmd . "\r\n");
print("debug");
$line = readResponse();
print("$line\n");

#close our socket
$sock->close();

#exit success if we make it here
exit 0;



#sub to find the </commandresponse> string that is found
#after setting result type to xml
sub readResponse {
 my $resp = "";
 my $line = "";

 $line = "";
 while(index($resp, "</commandresponse") < 0){
     $line = readline $sock;
     $resp = $resp . $line;
     #print("$line\n");
    }
     return $resp;
}


sub usage {

 print "\n\n\*****************************************************************************\n\n";
 print "Use: clui_ssl_sample.pl <host> <port> <username> <password> <command>\n\n";
print "        host - ip or name\n";
 print "        port - the port the clui is on\n";
 print "        username - admin user name\n";
 print "        password - admin password\n";
      print "        command - The command to send to the server (in quotes)\n\n";
```

```
    print "***********************************************************************\n";

} # end sub usage
```

CLUI architecture

The CLUI is composed of several parts:

- Client—The user-written client or the commands issued by the user. The Client communicates with the CLUI Listener.

- CLUI Server—Starts the CLUI framework. The CLUI Server reads the configuration file, which describes the available ports. Typically, there is one SSL (secure) and one non-SSL (unsecure) port. Each port allows a maximum number of sessions, which is specified in the configuration file. The CLUI Server starts a CLUI Listener for each configured port.

- CLUI Listener—Accepts new client connections. When a new connection is created, the CLUI Listener creates a new CLUI client context object. Each client establishes its own authentication.

- CLUI client context—Provides the context for a client session. Each client has a client context object. When a client session ends, the CLUI context object is destroyed.

  The CLUI client context checks the validity of all commands issued by the user. If a user command is QUIT, LOGIN, or SET OPTION, the client context processes the command directly. Otherwise, the CLUI client context passes the valid command to the CLUI DIRECTOR.

  When a command completes, the CLUI client context transforms the XML response into the specified result format. The default response format for a Telnet session is plain text. All other sessions have an XML default format.

- CLUI Director—Determines whether a command is a help request or a valid command. If the command is ?, *action* ?, *HELP*, or *HELP action*, the CLUI Director displays the types of commands that are supported. The CLUI Director routes the command to the appropriate CLUI Handler. If there is no appropriate CLUI Handler, the CLUI Director returns an error to the client.

- CLUI Handler—Performs the command. After the command completes, the CLUI Handler returns an XML response.

- Command parser—Parses the SSSU-stye command syntax. The parser fully qualifies all actions, targets, and attributes. Therefore, if you enter abbreviations or synonyms in a command, the parser replaces these with the full command names.

- XML command response—All command responses are in XML and must follow a consistent layout.

- Result code—Contains static result values. This enforces consistent result codes throughout the CLUI framework.

2 Command descriptions

This chapter contains the command descriptions of all of the commands in the HP Replication Solutions Manager CLUI.

ADD DR_GROUP

Synopsis

```
a[dd] dr[_group]|drg <dr_group name>
{destination_s[ystem]|ds={destination system}} vd[isk]={vdisk name}}
[acc[essmode]|am={none|noread|readonly}]
[com[ment]=<comment>]
[destination_d[isk_group]|ddg=<destination_disk_group>]
[destination_vd[isk_name]|dvn=<destination vdisk name>]
[destination_vr[aid_level]|dvrl={raid0|raid1|raid5}]
[fails[afe]|
[log_disk_group|ldg=<log_disk_group name>]
[nof[ailsafe]|nfs]
[nos[uspend]|ns]
[sus[pend]]
[wr[itemode]|wm={sync|synchronous|async|asynchronous}]
```

Description

The `ADD DR_GROUP` command creates a data replication group.

Switches

dr_group name	Specifies the name of the DR group you are creating.
destination_system	**Required.** Specifies the system where the replicated destination Vdisks are created. There is no default value.
vdisk	**Required.** Specifies the name of the source Vdisk to be added to the new DR group. This Vdisk is replicated to the destination DR group. There is no default value.
accessmode	Specifies the destination access mode for this DR group. Valid values include *none* (default), *noread*, or *readonly*.
comment	Adds a comment to the DR group. Use quotation marks to enclose the text of your comment. The maximum number of characters is 64.
destination_disk_group	Specifies the destination disk group for the destination Vdisk. The default disk group is the default value.
destination_vdisk_name	Specifies the name of the replicated destination Vdisk that is created when the source is added to the DR group. The default is to use the same name as the source Vdisk, unless there is a naming conflict on the destination array. In such a case, the destination array generates a Vdisk name.
destination_vraid_level	Specifies the destination Vraid level to use for the destination Vdisk. The default is RAID0.

failsafe	All write operations are immediately cancelled when the connection between the source and destination system fails. Failsafe is disabled (*nofailsafe*) by default.
log_disk_group	Specifies the log disk group for the DR group. If the name of the log disk group contains spaces, use quotation marks.
nofailsafe	All write operations are logged for later synchronization between the source and destination DR groups when the connection between the source and destination system fails. Failsafe is disabled (*nofailsafe*) by default.
nosuspend	Resumes the replication of data from the source to the destination, similar to a restored connection. Suspend is disabled (*nosuspend*) by default.
suspend	Pauses the replication of data from the source to the destination, similar to a failed connection. Suspend is disabled (*nosuspend*) by default.
writemode	Specifies the I/O interaction between the source and destination DR group. The possible values are *synchronous* or *asynchronous*. The default value is *synchronous*.

Example

```
add dr_group bk3888 destination_system=s2333 vdisk=corporatephoto
accessmode=readonly ldg=r4445o writemode-synchronous
```

ADD HOST_AGENT

Synopsis

```
a[dd] host_a[gent]|ha<host agent name>
```

Description

The `ADD HOST_AGENT` command adds a host agent.

Switches

host agent name	Specifies the name of the new host agent.

Example

```
add host_agent newhost
```

ADD MANAGED_SET

Synopsis

```
a[dd] managed[_set]|ms|mset <managed_set name>
{dr_group|drg
|host_a[gent]|ha
|host_v[olume]|hv
|s[torage_system]|ss
|vd[isk]}
```

Description

The `ADD MANAGED_SET` command creates a managed set. A managed set is a user-defined collection of resources bound together for management purposes. A managed set can contain DR groups, host agents, host volumes, storage systems, and virtual disks. All members in a single managed set must be of the same type.

Switches

managed_set name	Specifies the name of the managed set you want to create.
dr_group	Contains DR.
host_agent	Creates a managed set of host agents.
host_volume	Creates a managed set of host volumes.
storage_system	Creates a managed set of arrays.
Vdisk	Creates a managed set of Vdisks.

ADD CONTAINER

Synopsis

```
a[dd] cont[ainer]<container name>

{{s[ize]={size(gb)}}
|{disk_g[roup]|dg={disk group name}}
|[vraid_level|vrl={raid0|raid1|raid5}]}

|{{vd[isk]={vdisk name}}
|[disk_g[roup]|dg=<disk group name>]}
```

Description

Use the `ADD CONTAINER` command to create an empty Vdisk container that cannot be presented to a host. A container Vdisk can be used as pre-allocated storage space within which to create a fully-allocated snapshot.

Switches

container name	Specifies the name of the container Vdisk.
size	Specifies the container size in GB.
disk_group	Specifies the container's disk group. The disk group can be specified in UNC form or as the name found on the selected array.
VRaid_level	Specifies the container's VRaid level. The default is *VRAID0*.
Vdisk	Specifies the container's Vdisk. The resulting container has identical size, RAID, and storage pool attributes.

ADD SNAPCLONE

Synopsis

```
a[dd] snapc[lone]|sc<snapclone name>
{vd[isk]={vdisk name}}
```

```
[nowait]
[wait]

[[diskg[roup]dg=<diskgroup name>]
|[redundancy_level|rl={same|raid0|raid1|raid5}]]
```

Description

Use the `ADD SNAPCLONE` command to create a local replica of the specified Vdisk object.

Switches

snapclone name	Specifies the name of the snapclone.
nowait	Creates the snapclone without waiting for normalization to complete.
wait	Waits for the snapclone to normalize before returning to the prompt. This is the default. For large cloning operations, this may take some time to complete.
diskgroup	Specifies the disk group for this snapclone to be created in.
redundancy_level	Specifies which snapclone redundancy level to use. Valid values are `SAME`, `RAID0`, `RAID1`, or `RAID5`. The default is `SAME`. See the HP Business Copy administrator guide to determine the proper source and clone RAID combinations.

ADD SNAPSHOT

Synopsis

```
a[dd] snaps[hot]|ss<snapshot name>
{vd[isk]={vdisk name}}
[nowait]
[wait]

[[redundancy_level|rl={same|raid0|raid1|raid5}]
|[snapshot_type|st={demand_allocated|da|fully_allocated|fa}]]
```

Description

Use the `ADD SNAPSHOT` command to create a local replica of the specified Vdisk object.

Switches

snapshot name	Specifies the name of the snapshot.
nowait	Creates the snapshot without waiting for normalization to complete.
wait	Waits for the snapshot to normalize before returning to the prompt. This is the default.
redundancy level	Specifies the snapshot redundancy level to use. Valid values are `SAME`, `RAID0`, `RAID1`, or `RAID5`. The default is `SAME`.
snapshot_type	Specifies the snapshot type to use. Valid values are `space_efficient` or `fully_allocated`. The default is `demand_allocated`.

CAPTURE CONFIG_DATA

Synopsis

```
c[apture] c[onfig_data]|cfg
```

Description

The `CAPTURE CONFIG_DATA` command captures configuration data from the selected system. The displayed information can be used to recreate the system in the event of a failure. Jobs are not included in the configuration data. A scripting language is needed in conjunction with the command to view the data.

CAPTURE SYSTEM_DATA

Synopsis

```
c[apture] s[ystem_data]|sys
```

Description

The `CAPTURE SYSTEM_DATA` command allows you to capture the system configuration data. A scripting language is needed in conjunction with the command to view the data.

DELETE DR_GROUP

Synopsis

```
de[lete] dr[_group]|drg <dr_group name>
[delete|del]
```

Description

The `DELETE DR_GROUP` command removes specified DR groups from the array. If not specified, the DR groups are detached.

Switches

dr_group name	Specifies the name of the DR group that you want to remove from the storage system.
delete	Deletes the destination Vdisk(s). If you do not include this option in the command line, the array detaches the Vdisks from the DR group.

Example

```
de drg 5tt444 delete
```

DELETE HOST_AGENT

Synopsis

```
de[lete] host_a[gent]|ha<host agent name>
```

Description

The `DELETE HOST_AGENT` command deletes the host agent.

Switches

host agent name   Specifies the name of the host agent you want to delete.

Example

```
de host_a newhost123
```

DELETE JOB

Synopsis

```
de[lete] job <job name>
```

Description

The `DELETE JOB` command deletes jobs. You cannot delete jobs that have instances that are running or paused.

DELETE VDISK

Synopsis

```
del[ete] vd[isk]<vdisk name>
```

Description

The `DELETE VDISK` command deletes the specified Vdisk.

DELETE MANAGED_SET

Synopsis

```
de[lete] managed[_set]|ms|mset<managed_set name>
```

Description

The `DELETE MANAGED_SET` command deletes the specified managed set.

DELETE CONTAINER

Synopsis

```
del[ete] cont[ainer]<container name>
```

Description

The DELETE CONTAINER command deletes the specified container.

DELETE SNAPCLONE

Synopsis

```
del[ete] snapc[lone]<snapclone name>
```

Description

The DELETE SNAPCLONE command deletes the specified snapclone.

DELETE SNAPSHOT

Synopsis

```
del[ete] snaps[hot]<snapshot name>
```

Description

The DELETE SNAPSHOT command deletes the specified snapshot.

EXIT

Synopsis

```
exit
```

Description

The EXIT command ends and exits a CLUI session.

HELP

Synopsis

```
h[elp] <command name>
```

Description

The HELP command displays help for CLUI commands.

LOGIN

Synopsis

```
login
{username={username} |password={password}}
```

Description

The LOGIN command authenticates the user for the current management session.

Switches

username	**Required.**	Specifies the username to authenticate.
password	**Required.**	Specifies the password for the username.

SELECT HOST_AGENT

Synopsis

```
sel[ect] host[_agent]<host agent>
```

Description

The SELECT HOST_AGENT command selects the host agent.

Switches

host agent   Specifies the name of the host agent you are selecting.

Example

```
select host_agent pc34444
```

SELECT SYSTEM

Synopsis

```
sel[ect] sys[tem]<storage system name>
```

Description

The SELECT SYSTEM command selects the storage system.

Switches

storage system name   Specifies the name of the storage system that you want
                                       to manage.

Example

```
sel sys corporatephotolibrary
```

SET CLIENT

Synopsis

```
set client
{res[ult_format]|rf={block_text|csv|result_code|table_text|xml}}
```

Description

The SET CLIENT command specifies the result format for the CLUI client. The client can receive the results of a command as block text, CSV, table text, or XML.

Switches

result_format   Specifies the result format. The options include *block_text*, *csv*,
                                 result_code, *table_text*, and *xml*.

Examples

Block text format

```
NY>show ms full
0 Success
Name........:manset1
Type........:VirtualDisk
Comment.....:
Dae Created 6/14/04 4:39 PM
Member Count: 0

Name........:manset2
Type........:Connection
Comment.....:
Date Created: 6/14/04 4:39 PM
Member Count: 0
```

CSV format

```
NY>show ms full
RC=0 Success
Name,Type,Comment,Date Created,Member Count,
manset1,VirtualDisk,,6/14/04 4:39 PM,0,
manset2,Connection,,6/14/04 4:39 PM,0,
```

Result code format

```
NY>show ms full
RC=0 Success
```

Table text format

```
NY>show ms full
0 Success
```

```
Name      Type         Comment Date Created      Member Count
-------   -----------  ------- ---------------   ------------
manset1 VirtualDisk            6/14/04 4:39 PM 0
manset2 Connection            6/14/04 4:39 PM 0
```

XML format

```
NY>sho ms full
<?xml version="1.0" encoding="UTF-8"?>
<commandresponse>
 <resultcode>0 Success</resultcode>
 <command>SHOW MANAGED_SET FULL</command>
 <desciption>Managed set SHOW</description>
 <table>
  <heading>
   <column>Name</column>
   <column>Type</column>
   <column>Date created</column>
   <column>Member Count</column>
  </heading>
  <row>
   <column>manset1</columm
   <column>VirtualDisk</column>
   <column>6/14/04 4:39 PM</column>
   <column>0</column>
  </row>
  <row>
   <column>manset2</column>
   <column>Cnnection</column>
   <column>6/14/04 4:39 PM</column>
   <column>0</column>
  </row>
 </table>
</commandresponse>
```

SET DR_GROUP

Synopsis

```
set dr[_group]|drg <dr_group name>
[acc[essmode]|am={none|noread|readonly}]
[autos[uspend]]
[com[ment]=<comment>]
[failo[ver]|fo]
[fails[afe]|fs]
[force_f[ull_copy]|ffc]
[home]
[max_log_disk_size|mlds=<maximum log disk size (mb)>]
[na[me]=<new_dr_group name>]
[noautos[uspend]]
[nof[ailsafe]|nfs]
[nos[uspend]|ns]
[refresh]
[sus[pend]]
[suspend_and_failover|saf]
[wr[itemode]|wm={sync|synchronous|async|asynchronous}]
[[add[_vdisk]|av=<vdisk name>]
|[destination_vd[isk_name]|dvn=<destination vdisk name>]
|[destination_vr[aid_level]|dvrl={vraid0|vraid1|vraid5|same}]]
|
[{rem[ove_vdisk]|remvd|rvd=<vdisk name>}}
[del[ete]]
```

Description

The `SET DR_GROUP` command modifies the properties of a DR group.

Switches

dr_group name	Specifies the name of the DR group you are modifying.
accessmode	Specifies the destination access mode for the DR group. The values are *none* (default), *noread*, or *readonly*.
autosuspend	Sets the DR group to *autosuspend* on link down mode (suspend on failover).
add_vdisk	Specifies the name of the source Vdisk that you are adding to the DR group. When you use the *add_vdisk* switch, the software automatically creates the destination Vdisk in the destination storage system. When you add a Vdisk, you can specify two additional switches: *destination_disk_group* and *destination_vdisk_name*.
comment	Adds a comment to the DR group. Enclose the comment text in quotation marks if there are spaces in the comment. The maximum number of characters is 64.
delete	Removes the destination Vdisk. If you do not specify the `remove_vdisk` switch, the software detaches the Vdisk from the DR group; however, the Vdisk remains in the array.
failover	Reverses the roles of the DR group. The source becomes the destination, and the destination becomes the source.
failsafe	Halts all write operation, if the connection between the source and destination arrays fail.
force_full_copy	De-allocates the group write log and forces each member to do a complete copy of its contents to its remote partner.
home	Sets the DR group to function as the "home" side of the replication set.
max_log_disk_size	Specifies the maximum log disk size for the source DR group in MB.
name	Specifies the name of the DR group.
noautosuspend	Sets the DR group to *noautosuspend* on link down mode (no auto suspend on failover).
nofailsafe	Logs all write operations for later synchronization if the connection between the source and destination arrays fail.
nosuspend	Resumes the replication of data from the source to the destination, similar to a restored connection.
refresh	Performs a manual refresh of the DR group properties prior to other operations that may have been specified in this command.
remove_vdisk	Specifies the name of the source Vdisk you want removed from the DR group. The destination Vdisk is removed from the DR group, but it is not deleted. The destination Vdisk remains as an independent Vdisk. Use the *delete* switch to permanently delete the remote Vdisk from the array.
suspend	Pauses data replication from the source to the destination, similar to a failed connection.
suspend_and_failover	Pauses data replication from the source to the destination, then reverses the roles of the DR group. The source becomes the destination and the destination becomes the source.

writemode	Indicates the I/O interaction between the source and destination DR group. The possible values are *synchronous* and *asynchronous*.
destination_vdisk_name	Specifies the replicated destination Vdisk name that is created when the source is added to the DR group. The default has the same name as the source unless there is a name conflict. In such a case, the destination name will be auto-generated by the array.
destination_vraid_level	Specifies the destination VRaid level to use for the destination Vdisk. Valid only when used with `add_vdisk` switch. Valid values include *VRAID0*, *VRAID1*, *VRAID5*, or *SAME*. The default is *SAME*.

Example

```
set dr_group group455 name=gr100 wm=asynchronous
```

SET HOST_AGENT

Synopsis

```
set host_a[gent]|ha<host agent name>}
[com[ment]]=<a comment>
[mount_v[olume]|mv=<mount volume>]
[re[scan]
[refresh]
[ru[n]=<command>]

{mount_p[oint]|mp={mount point}}
[unmount_v[olume]|uv=<volume name>]
```

Description

The `SET HOST_AGENT` command modifies the properties of the host agent.

Switches

host agent name	Specifies the name of the host agent you want to modify.
comment	Adds a comment to the host agent.
mount_volume	Specifies the host volume to mount on the host. When you use this switch, the *mount_point* switch is required.
rescan	Rescans the host agent for changes.
run	Specifies a command to run on the host. The returned information includes the result code from the host agent, the command's result code, the host agent's result, the command's result, the system error, and the system out information for the command.
mount_point	Specifies the mount point. Use this option only if you use the *mount_volume* or *unmount_volume* switches.
unmount_volume	Unmount the storage from the host. If you use this option, the *mount_point* switch is required.

SET JOB

Synopsis

```
set job<job name or instance name>
{[ab[ort]]
|[cont[inue]]
|[des[cription]=<new description>]
|[disable_s[chedule]|ds=<scheduleid>]
|[enable_s[chedule]|es=<scheduleid>]
|[name=<new name>]
|[pause]}
|{{run}
 |[mode={validate|normal|skip_validation}]
 |[nowait]
 |[wait]}
```

Description

The `SET JOB` command modifies the properties of a job.

Switches

abort	Stops the job operation.
con-tinue	Resumes the job instance.
de-scrip-tion	Changes the job's description.
dis-able_sched-ule	Disables the job schedule using the supplied *scheduleid*.
en-able_sched-ule	Enables the job schedule using the supplied *scheduleid*.
name	Changes the name of the job.
pause	Pauses the job instance.
run	Runs the job.
mode	Sets the run mode for the selected job. Use this switch with the *run* switch to set the *run* mode to *normal*, *validate*, or *skip validation*. The default value is *normal*.
nowait	Launches the job without waiting for a job to complete.
wait	Waits for the job to complete before returning the command prompt. This is the default.

SET MANAGED_SET

Synopsis

```
set managed[_set]|ms|mset<managed_set name>
[a[ddmember]|am=<member name>]
[com[ment]=<comment>]
[failo[ver]|fo]
[fails[afe]|fs]
[h[ost]=<host name>]
```

```
[na[me]=<name>]
[nof[ailsafe]|nfs]
[nos[uspend]|ns]
[rem[ovemember]|rm=<member name>]
[sus[pend]]
```

Description

The `SET MANAGED_SET` command modifies the properties of a managed set.

Switches

managed_set name	Specifies the name of the managed set you want to modify.
addmember	Adds a member to the managed set. The name must correspond to an object of the same type that is contained in the managed set. For example, you can add a DR group only to a DR group managed set. Use quotation marks if the name contains spaces.
comment	Modifies the comment text for a managed set.
failover	Commands the interface to failover all members of the managed set.
failsafe	Enables failsafe mode for all members of a managed set.
host	Specifies the host name where the host volume exists. This switch is required if the *addmember* switch is used.
name	Renames the managed set. Use quotation marks if the name contains spaces.
nofailsafe	Disables the failsafe mode for all of the members of a managed set.
nosuspend	Resumes replication for all members in the managed set.
removemember	Removes a member from the managed set. Use quotation marks if the name contains spaces.
suspend	Suspends replication on all members in the managed set.

SET SYSTEM

Synopsis

```
set sys[tem]<system name>
{[refresh]
|[refresh_all]}
```

Description

The `SET SYSTEM` command displays the properties of the system.

Switches

system name	Specifies the name of the system.
refresh	Refreshes the data for the array. If the name is omitted, the selected array will be refreshed.
refresh_all	Refreshes all of the array data for the entire replication server.

SET VDISK

Synopsis

```
set vd[isk]<vdisk name>
[cache_m[ode]|cm={write_t[hrough]|wt|write_b[ack]|wb}]
[remove_p[resentation]|rp=<host name>]
[refresh]

[{add_p[resentation]|ap={host name}}
|[lun=<lun number>]]

[[inst[ant_restore]|instrest|irestore=<disk name>]]
```

Description

Use the SET VDISK command to modify Vdisk properties.

Switches

cache_mode   Sets the Vdisk's cache mode to write-through or write-back.

> △ **CAUTION:**
> Leaving a Vdisk in write-through mode will cause a performance degradation
> for that Vdisk.

vdisk name   Specifies the name of the Vdisk.

remove_presenta-   Selects an EVA host name to which to unpresent your Vdisk.
tion

add_presentation   Selects an EVA host name to which to present your Vdisks. This allows those
                      hosts access to Vdisk data. Presenting more than one host to a volume
                      enables possible write conflicts. EVA hosts are port lists and they may consist
                      of ports from one or more physical host arrays. Presenting to such an EVA
                      host enables possible write conflicts.

lun   A valid LUN number for this host. The default is auto select.

refresh   Performs a manual refresh of the Vdisk properties prior to other operations
                      that may have been specified in this command.

instant_restore   Provides a disk name to perform a synchronize operation. The disk provided
                      in this set command will be synchronized with the disk named with the
                      instant_restore switch.

SET CONTAINER

Synopsis

```
set cont[ainer]<container name>

[pre[allocated_snapclone_vdisk]|psvd=<vdisk name>]
[refresh]
```

Description

Use the SET CONTAINER command to modify the container's properties.

Switches

container name	Specifies the name of the container.
preallocated_snap-clone_vdisk	Specifies the name of the Vdisk to use for a pre-allocated snap to this container.
refresh	Performs a manual refresh of the container properties prior to other operations that may have been specified in this command.

SHOW DR_GROUP

Synopsis

```
sho[w] dr[_group]|drg <dr_group name>
[f[ull]]
[l[ist]]
[m[embers]]
[refresh]
```

Description

The SHOW DR_GROUP command displays the properties of the DR group.

Switches

dr_group name	Specifies the name of the DR group.
full	Use this switch instead of the DR group name to show the properties of all data replication groups in the array.
list	Lists only the names of the DR groups displayed. The software does not display detailed information.
members	Displays the members of the DR group.
refresh	Performs a manual refresh of the DR group properties prior to other operations that may have been specified in this command.

Example

```
show dr_group full
```

SHOW HOST_AGENT

Synopsis

```
sho[w] host_agent|ha<host_agent name>
[cluster]
[f[ull]
[hba=<hba name>]
[hbas]
[host_volume|hv=<host volume name>]
[host_volumes|hvs]
[l[ist]]
[mount_point|mp=<mount point>]
[mount_points|mps]
```

Description

The SHOW HOST_AGENT command shows the properties of the host agent.

Switches

host agent name	Specifies the name of the host agent you want to view.
cluster	Shows the cluster information for the specified host agent.
full	If you use *full* instead of the *host agent name*, the interface shows the details of all the host agents.
hba	Shows information for the specified Host Bus Adapter.
hbas	Lists the Host Bus Adapters on the host rather than the default volume information.
host_volume	Shows information for the specified host volume on the host.
host_volumes	Shows host volumes for the specified host.
list	Displays the names of the host agents without detailed information.
mount_point	Shows information for the specified mount point on the host.
mount_points	Shows the mount points on the host instead of the default volume information.

Example

```
show host_agent full
```

SHOW HOST_VOLUME

Synopsis

```
sho[w] host_vol[ume]|hostvol|hv<host_volume name>
[f[ull]]
[h[ost]=<host name>]
[l[ist]]
[m[ounts]]
[vd[isks]]
```

Description

The SHOW HOST_VOLUME command displays the host volume properties.

Switches

host volume name	Specifies the name of the host volume you want to view.
full	Use *full* instead of *host volume_name* to show details for all the host volumes.
host	**Required.** Specifies the host name where the host volume exists.
list	Lists the names of the host volumes without detailed information.
mounts	Shows all mounts for the specified host volume.
vdisks	Shows all Vdisks for the specified host volume.

SHOW JOBS

Synopsis

```
sho[w] job[s]<job name>
[events]
[f[ull]]
[i[nstances]]
[l[ist]]
[sched[ule]|scheduled]
[t[asks]]
```

Description

The SHOW JOB command displays the job's properties.

Switches

job name	Specifies the job for which to display information.
events	Displays the job instance event status information.
full	Shows details about all of the jobs. If you use this switch, you do not have to specify the *job name*.
instances	Shows instance information for each job run.
list	Displays only the names of the jobs without detailed information.
schedule	Displays the job schedule information.
tasks	Displays the complete listing of the job task steps.

SHOW MANAGED_SET

Synopsis

```
sho[w] managed[_set}|ms|mset<managed_set name>
[f[ull]]
[l[ist]]
[m[embers]]
```

Description

The SHOW MANAGED_SET command shows the properties of a managed set.

Switches

managed_set name	Specifies the name of the managed set for which you want to see details.
full	If this switch is used instead of the managed set name, the interface shows the details of all the managed sets.
list	Displays only the names of the managed sets without the detailed information.
members	Lists the members of the managed set, not the properties of the managed set.

SHOW SYSTEM

Synopsis

```
sho[w] sys[tem]<system name>
[f[ull]]
[l[ist]]
[man[aged_set_member]|ms_member|msm]
```

Description

The SHOW SYSTEM command shows the selected system's properties.

Switches

system name	Specifies the system for which to display information.
disk_groups	Displays information about the selected array's disk groups.
full	If you use *full* instead of the *system name*, the interface shows details for all of the systems.
list	Displays the names of the systems without detailed information. If *full* or the *system name* is excluded, *list* is the default.
managed_set_member	Shows information for the managed sets that each array is a member of.

SHOW VDISK

Synopsis

```
sho[w] vd[isk]<vdisk name>

[f[ull]]
[l[ist]]
[man[aged_set_member]|ms_member|msm]
[perf[ormance]]
[pres[entation]]
[refresh]
[repl[ication]]
```

Description

The SHOW VDISK command shows the selected array's properties.

Switches

vdisk name	Specifies the Vdisk for which to display information.
full	If you use *full* instead of the *Vdisk name*, the interface shows all of the Vdisks.
list	Displays the names of the Vdisks without the detailed information. If *full* or the *Vdisk name* is excluded, *list* is the default.
managed_set_member	Shows information for the managed sets that each Vdisk is a member of.
performance	Shows performance attributes for the Vdisk.
presentation	Shows presentation attributes for the Vdisk.
refresh	Performs a manual refresh of the Vdisk properties prior to other operations that may have been specified in this command.
replication	Shows replication attributes for the Vdisk.

SHOW CONTAINER

Synopsis

```
sho[w] cont[ainer]<container name>

[f[ull]]
[l[ist]]
[man[aged_set_member]|ms_member|msm]
[perf[ormance]]
[pres[entation]]
[refresh]
[repl[ication]]
```

Description

The SHOW CONTAINER command displays the selected container properties.

Switches

container name	Specifies the name of the container.
full	Shows the details of all of the containers.
list	Displays the names of the containers without the detailed information. If *full* or the container name is excluded, *list* is the default.
managed_set_member	Displays all members of the specified managed set.
performance	Displays the performance properties for the specified container.
presentation	Displays the presentation properties for the specified container.
refresh	Performs a manual refresh of the container properties prior to other operations that may have been specified in this command.
replication	Displays the replication properties for the specified container.

SHOW SNAPCLONE

Synopsis

```
sho[w] snapc[lone]<snapclone name>

[f[ull]
[l[ist]]
[man[aged_set_member]|ms_member|msm]
[perf[ormance]]
[pres[entation]]
[refresh]
[repl[ication]]
```

Description

Use the SHOW SNAPCLONE command to display information regarding the array's Vdisks. It returns results identical to that of the SHOW DISK switch.

Switches

snapclone name	Specifies the name of the snapclone.
full	Shows the details of all snapclones.
list	Shows the names of the snapclone without the detailed information. If full or the the snapclone name is excluded, list is the default.
managed_set_member	Displays the managed sets each snapclone is a member of.
performance	Displays the performance properties for the snapclone.
presentation	Displays the presentation properties for the snapclone.
refresh	Performs a manual refresh of the snapclone properties prior to other operations that may have been specified in this command.
replication	Displays the replication properties for the snapclone.

SHOW SNAPSHOT

Synopsis

```
sho[w] snaps[hot]<snapshot name>

[f[ull]]
[l[ist]]
[man[aged_set_member]|ms_member|msm]
[perf[ormance]]
[pres[entation]]
[refresh]
[repl[ication]]
```

Description

Use the SHOW SNAPSHOT command to display the specified snapshot properties.

Switches

snapshot name	Specifies the name of the snapshot.

full	Shows the details of all snapshots.
list	Shows the names of the snapshot without detailed information. If `full` or the snapshot name is excluded, `list` is the default.
managed_set_member	Displays the managed sets each snapshot is a member of.
performance	Displays the performance properties of the snapshot.
presentation	Displays the presentation properties of the snapshot.
refresh	Performs a manual refresh of the snapshot properties prior to other operations that may have been specified in this command.
replication	Displays the replication properties of the snapshot.

A XML command response format

The XML command response object creates the XML response. The command response can contain multiple tables, rows, heading columns, and row columns. The heading columns are similar to tag/value pairs. For every column in the heading, you should include a column in each row. This will allow CSV and other output formats to properly format the data.

Here is a sample:

```
<?xml version="1.0" encoding="utf-8"?>

<commandresponse>
 <resultcode>value</resultcode>
 <command>original command</command>
 <description>description of command</description>
 <table>
  <heading>
   <column>heading column</column>
   <column>heading column</column>
  </heading>
  <row>
   <column>row column</column>
   <column>row column</column>
  </row>
  <zipfile>
   <bindata>
    <![CDATA[data]]>
   </bindata>
  </zipfile>
 </table>
</commandresponse>
```

B CLUI Handler XML configuration file

The CLUI Handler configuration file describes the CLUI Handler and supported commands to the CLUI framework. The configuration file must have a .cluihandlerxml extension. Here is a sample configuration file:

```
<cluihandler frameworkHelpEnabled="true">
 <codebase>com.hp.my.package.structure.MyCluiHandler</codebase>
 <priority>normal</priority>
 <command hidden="true">
  <action>A*DD
   <help>Add a managed set.</help>
  </action>

 <target>MANAGED*_SET
   <synonym>MS</synonym>
   <synonym>MSET</synonym>
   <targetValidValue>Managed_Set_Name</targetValidValue>
   <help>Use the ADD MANAGED_SET command to create a logical group
 that contains specific entries based on the type of the managed set specified.</help>
</target>

<exclusiveGroup required="true">
  <switch required="true">DR_GROUP
  <synonym>DRG</synonym>
  <help>Contains data replication group objects</help>
  </switch>

 <switch required="true">CON*NECTION
  <help>Contains connection objects</help>
  </switch>

 <switch required="true">HOST_A*GENT
  <synonym>HA</synonym>
  <help>Contains host objects</help>
 </switch>

 <switch required="true">HOST_V*OLUME
  <synonym>HV</synonym>
  <help>Contains host volume objects</help>
  </switch>

<switch required="true">S*TORAGE_SYSTEM
  <synonym>SS</synonym>
  <help>Contains storage system objects</help>
  </switch>

 <switch required="true">VD*ISK
  <help>Contains virtual disk objects</help>
  </switch>
 </exclusiveGroup>


</command></cluihandler>
```

NOTE:
Synonyms and help can be added and modified, but actions and targets cannot be modified.

C BC 2.x EVMCL to RSM 1.x job-related commands

BC 2.x EVMCL full syntax

```
evmcl <bc_server> <bc_command> <bc_job> [/output=Filename.ext][/i]
```

RSM 1.x syntax

```
set job <job name or instance name>
|[cont[inue]]
|[des[cription]=<new description>]
|[disable_s[chedule]|ds=<scheduleid>]
|[name=<new name>]
|[pause]}
 |[mode={validate|normal|skip_validation}]
 | [nowait]

sho[w] job[s] <job name>
[f[ull]]
[i[nstances]]
[sched[ule]|scheduled]
[t[asks]]
```

Comparable BC 2.x EVMCL and RSM 1_x commands

Table 2 Comparable BC 2.x EVMCL and RSM 1_x commands

BC 2.x EVMCL commands	RSM 1.x CLUI commands	RSM 1.x CLUI switches
evmcl *<bc server>* abort *<job name>*	Set job *<job instance name>*	abort
evmcl *<bc server>* continue *<job name>*	Set job *<job instance name>*	continue [wait|nowait] (Default is `wait`)
evmcl *<bc server>* pause *<job name>*	Set job *<job instance name>*	pause
evmcl *<bc server>* run *<job name>*	Set job *<job name>*	run [wait|nowait] (Default is `nowait`)
evmcl *<bc server>* status *<job name>*	Show job *<job instance name>*	
evmcl *<bc server>* statusdetail *<job name>*	Show job *<job instance name>*	events
evmcl *<bc server>* statusfull *<job name>*		
evmcl *<bc server>* undo *<job name>*		
evmcl *<bc server>* validate *<job name>*	Set job *<job name>*	run mode=validate
evmcl *<bc server>* getjoblist	Show job	list (list of job names)
evmcl *<bc server>* getjoblist	Show job	full (list of jobs and properties)
	Show job	(list of single job properties)
	Show job	tasks (list of single job's tasks)

Table 3 RSM Return code values for all commands

Return codes	Commands
0	SUCCESS
500	COMMAND_FAILURE
501	COMMAND_PROCESS_FAILURE
502	COMMAND_PROCESS_PARTIAL_FAILURE
503	FAILURE_COMMAND_PROCESSOR
504	INVALID_COMMAND_SYNTAX
505	INVALID_SIGNON
506	SECURITY_VIOLATION
507	UNKNOWN_COMMAND
508	JOB_PAUSED_ERROR
509	JOB_PAUSED_NORMAL

Index